REMARKS

Claims 1-20 are currently pending in the application. Claims 1, 9 and 20 are independent claims. Reconsideration and withdrawal of all pending rejections in view of the above amendments and following remarks is respectfully requested.

35 U.S.C. § 103 Rejections

Over Sarukkai with Glance

Claims 1 and 4-11 (and presumably also claims 12, 15, 16, 18 and 19) are rejected under 35 U.S.C. § 103(a) as being unpatentable over U. S. Patent No. 6,775,695 to SARUKKAI in view of U. S. Patent No. 6,415,368 to GLANCE et al. This rejection is respectfully traversed.

Claims 1 and 9 are directed to a method for adapting to change in a demand on a web server. In particular, representative claims 1 and 9 recite, in pertinent part:

> associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers; and
>
> analyzing the identifications of web pages requested by the browsers to determine caching priorities for the web server.

Claim 9 further recites:

> altering a server cache responsive to the caching priorities.

Such features are not disclosed or suggested by the combination of SARUKKAI and GLANCE.

Applicants do not dispute that SARUKKAI relates to document caching with a server (see col. 1, lines 6-9). Nor do Applicants dispute that the disclosed system is capable of "monitoring the number of documents requested by a client in a current session, placing a document requested by the client in a file cache according to a caching algorithm that is based, at least in part, on the number of documents requested by the client in the current session, and accessing the document in the file cache when the document is requested subsequently by the client. The wide area network is typically the Internet" (see col. 2, lines 14-21). However, it is clear from a fair reading of

this document that SARUKKAI stores the document itself into the cache based on the probability that it will be requested (see col. 8, lines 32-43). In contrast, the invention provides for associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers. As the Examiner will note from paragraph [0014] of the instant published application 2002/0165909, session tracking objects constitute information about the requests for web pages by the browsers and this information identifies each of the browsers. This distinction is not without a difference because the instant invention enables session tracking of the session objects in order to detect changes in demand more rapidly.

GLANCE does not cure the deficiencies of SARUKKAI. GLANCE merely discloses a system and method of caching based on a recommender system. The disclosed system employees a democratic caching generally shown by reference numeral 10. A recommender system 16 provides value information pertaining to items to be stored in cache 24 based on user input (col. 4, liens 43-53) that includes implicit site recommendations (col. 5, lines 24-55) and explicit URL recommendations (col. 5, lines 65 et seq.). GLANCE, like SARUKKAI, does not disclose or suggest associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers. GLANCE does not even determine caching priorities for the server by analyzing the identifications of web pages requested by the browsers.

As GLANCE fails to cure the deficiencies of SARUKKAI, GLANCE cannot serve to provide the motivation to combine these references. Furthermore, even if SARUKKAI and GLANCE were properly combinable, the combination would not result in the invention as recited in at least claims 1 and 9 including, *inter alia*, analyzing the identifications of web pages requested by the browsers to determine caching priorities for the web server.

Because, there is no suggestion or disclosure in SARUKKAI and GLANCE separately or in any proper combination that render obvious the features of the present claimed invention, the Examiner is respectfully requested to withdraw the rejection under 35 U.S.C. § 103.

Thus, claims 1 and 9 are allowable over the combination of SARUKKAI and GLANCE. Moreover, claims 4-8, 10 -12, 15, 16, 18 and 19 depend from claims 1 and 9, and are also allowable for the same reasons as claims 1 and 9, as well as for their added features.

Over Sarukkai with Glance and Ronald

Claims 2 and 3 are rejected under 35 U.S.C. § 103(a) as being unpatentable over SARUKKAI in view of GLANCE, and further in view of U. S. Patent Application Publication No. 2003/0041143 to RONALD et al. This rejection is respectfully traversed.

As explained above, SARUKKAI relates to document caching with a server (see col. 1, lines 6-9) and discloses a system is capable of "monitoring the number of documents requested by a client in a current session, placing a document requested by the client in a file cache according to a caching algorithm that is based, at least in part, on the number of documents requested by the client in the current session, and accessing the document in the file cache when the document is requested subsequently by the client. The wide area network is typically the Internet" (see col. 2, lines 14-21). However, it is clear from a fair reading of this document that SARUKKAI stores the document itself into the cache based on the probability that it will be requested (see col. 8, lines 32-43). In contrast, the invention provides for associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers.

Again, GLANCE does not cure the deficiencies of SARUKKAI. GLANCE merely discloses a system and method of caching based on a recommender system. As noted above, GLANCE, like SARUKKAI, does not disclose or suggest associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers. GLANCE does not even determine caching priorities for the server by analyzing the identifications of web pages requested by the browsers.

RONALD does not cure the deficiencies of SARUKKAI and GLANCE at least because it also does not disclose or suggest the features of claim 1, from which claims 2 and 3 depend. RONALD relates to a system for obtaining demographic information

about network users. There is no apparent disclosure with regard to associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers. Nor has the Examiner even alleged as much.

Thus, RONALD fails to cure the deficiencies of SARUKKAI and GLANCE. Furthermore, even if SARUKKAI, GLANCE and RONALD were properly combinable, the combination would not result in the invention as recited in at least claim 1 including, *inter alia*, analyzing the identifications of web pages requested by the browsers to determine caching priorities for the web server.

Because, there is no suggestion or disclosure in SARUKKAI, GLANCE and RONALD separately or in any proper combination that render obvious the features of the present claimed invention, the Examiner is respectfully requested to withdraw the rejection under 35 U.S.C. § 103.

Thus, claim 1 is allowable over the combination of SARUKKAI, GLANCE and RONALD. Moreover, claims 2-3 depend from claim 1, and are also allowable for the same reasons as claim 1, as well as for their added features.

Over Sarukkai with Glance and Klopp Lemon

Claims 13, 14 and 17 are rejected under 35 U.S.C. § 103(a) as being unpatentable over SARUKKAI in view of GLANCE, and further in view of U. S. Patent Application Publication No. 2002/0156881 to KLOPP LEMON et al. This rejection is respectfully traversed.

Again, SARUKKAI relates to document caching with a server (see col. 1, lines 6-9) and discloses a system is capable of "monitoring the number of documents requested by a client in a current session, placing a document requested by the client in a file cache according to a caching algorithm that is based, at least in part, on the number of documents requested by the client in the current session, and accessing the document in the file cache when the document is requested subsequently by the client. The wide area network is typically the Internet" (see col. 2, lines 14-21). However, it is clear from a fair reading of this document that SARUKKAI stores the document itself into the cache based on the probability that it will be requested (see col. 8, lines 32-43).

In contrast, the invention provides for associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers.

Again, GLANCE does not cure the deficiencies of SARUKKAI. GLANCE merely discloses a system and method of caching based on a recommender system. As noted above, GLANCE, like SARUKKAI, does not disclose or suggest associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers. GLANCE does not even determine caching priorities for the server by analyzing the identifications of web pages requested by the browsers.

KLOPP LEMON does not cure the deficiencies of SARUKKAI and GLANCE at least because it also does not disclose or suggest the features of claims 1 and 9, from which the above-noted claims depend. While it is true that KLOPP LEMON relates to a system for monitoring HTTP transactions between a server and a client and that the disclosed system uses servlets (see Fig. 1), there is no apparent disclosure with regard to associating session tracking objects with browsers that access a web server, wherein the session tracking objects include identifications of web pages requested by the browsers. Nor has the Examiner even alleged as much.

Thus, KLOPP LEMON fails to cure the deficiencies of SARUKKAI and GLANCE. Furthermore, even if SARUKKAI, GLANCE and KLOPP LEMON were properly combinable, the combination would not result in the invention as recited in at least claim 1 including, *inter alia*, analyzing the identifications of web pages requested by the browsers to determine caching priorities for the web server.

Because, there is no suggestion or disclosure in SARUKKAI, GLANCE and KLOPP LEMON separately or in any proper combination that render obvious the features of the present claimed invention, the Examiner is respectfully requested to withdraw the rejection under 35 U.S.C. § 103.

Thus, claims 1 and 9 are allowable over the combination of SARUKKAI, GLANCE and KLOPP LEMON. Moreover, claims 13, 14 and 17 depend from claims 1 and 9, and are also allowable for the same reasons as claims 1 and 9, as well as for their added features.

Over Asai with Knouse and Glance

Claim 20 is rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,760,765 to ASAI et al. in view of U.S. Patent Application Publication No. 2003/0074580 to KNOUSE et al, and further in view of GLANCE. This rejection is respectfully traversed.

Claim 20 is directed to a method for adapting to change in a demand on a web server. In particular, representative claim 20 recites, in pertinent part:

> determining whether HTTP session objects exist for browsers, wherein the HTTP session objects enable session tracking;
> associating session tracking objects with the browsers that access a web server which includes a plurality of servlets, a caching algorithm, and a fast memory cache, wherein the session tracking objects include identifications of web pages requested by the browsers;
> if an HTTP session object does not exist for one of browsers which requested one of the web pages, creating with the web server an HTTP session object for the browser;
> analyzing the identifications of web pages requested by the browsers to determine caching priorities for the web server; and
> altering a server cache responsive to the caching priorities,
> wherein the method ensures that a web site adapts to changes in demand.

Such features are not disclosed or suggested by the combination of ASAI, KNOUSE and GLANCE.

Applicants do not dispute that ASAI relates to an access system application program interface (see abstract). However, the Examiner is not correct that this document relates to a method for adapting to change in demand on a web server. It is clear from a fair reading of this document that ASAI does not disclose or suggest the above noted steps.

The Examiner identifies the session management unit 11 in Figs 1, 5 and 10 as being equivalent to a method for adapting to change in demand on a web server. Conveniently lacking from this assertion, however, is the location of any language in ASAI which specifically teaches this feature.

The Examiner also identifies col. 19, lines 36-42 and col. 19, line 51 to col. 20, line 40 as disclosing the associating step. Applicants disagree. The noted language of ASAI merely states the following:

For example, if a stream request arrives at the cache server 10.sub.1 or 10.sub.2 for the streaming data 701 stored in the cache servers 10.sub.1 and 10.sub.2 in common, the session management tables 53.sub.2 and 53.sub.3 in FIG. 9 each register information about that request. The information registered in the session management tables 53.sub.2 and 53.sub.3 is updated as the streaming data is distributed.

FIG. 10 is a diagram showing the session management tables 53.sub.2 and 53.sub.3 shown in FIG. 9. In each of the session management tables 53.sub.2 and 53.sub.3, a session identifier for identifying the session and a packet identifier indicating a packet most recently sent out are registered. Assume herein that the maximum number MAX of registrable sessions in each of the session management tables 53.sub.1 to 53.sub.8 is 8.

In the session management tables 53.sub.2 and 53.sub.3, sessions are identified in a reverse order (symmetrically in a table field). That is, the session identified by a session identifier I (0.ltoreq.{character pullout} I<MAX) in the session management table 53.sub.2 is identified by a session identifier (MAX-I-1) in the session management table 53.sub.3. For example, packet identifiers "100" and "510" of the sessions identified by session identifiers "0" and "1" in the session management table 53.sub.2 are registered in the session management table 53.sub.3 as identified by session identifiers "7" and "6", respectively.

Table boundary values 54.sub.2 of the session management table 53.sub.2 and 54.sub.3 session management table 53.sub.3 indicate numbers of the session identifiers at which the information registered in the session management tables 53.sub.2 and 53.sub.3 is divided into two. As described above, in the session management tables 53.sub.2 and 53.sub.3, sessions are registered in the reverse order. Therefore, if the table boundary value of one session management table is set to F (0.ltoreq.{character pullout} F<MAX), the table boundary value of the other session management table is set to (MAX-F). In FIG. 10, the table boundary value 543 of the session management table 53.sub.3 is set to "3", while the table boundary value 54.sub.3 of the session management table 53.sub.3 is set to "5".

The data distribution units 12.sub.1 to 12.sub.4 of the cache servers 10.sub.1 to 10.sub.4 transmit to the terminal only streaming data with respect to a session with the session identifier I equal to or smaller than the table boundary value F. More specifically, the cache servers 10.sub.1 to 10.sub.4 repeat the following first and second steps.

First step: Compare the session identifier I with the table boundary value F. If I<F, extract, from the streaming data storage unit, a packet that immediately comes after the packet corresponding to the session identifier I in the session management table, and send the extracted packet to the terminal.

> Second step: Update the value of the packet identifier corresponding to the session identifier I in the session management table to the value of the packet identifier of the packet sent out in the above first step.
>
> For example, in FIG. 10, for the sessions identified by the session identifiers "0" to "2" in the session management table 53.sub.2 of the cache server 10.sub.1 (surrounded by a thick line in FIG. 10 on the left), the data distribution unit 121 of the cache server 10.sub.1 transmits the relevant streaming data and updates the values of the packet identifiers. Similarly, for the sessions identified by the session identifiers "0" to "4" in the session management table 53.sub.3 of the cache server 10.sub.2 (surrounded by a thick line in FIG. 10 on the right), the data distribution unit 12.sub.2 of the cache server 10.sub.2 transmits the relevant streaming data and updates the values of the packet identifiers.

Applicants are at a loss to understand how such language can be interpreted to disclose associating session tracking objects with the browsers that access a web server which includes a plurality of servlets, a caching algorithm, and a fast memory cache, wherein the session tracking objects include identifications of web pages requested by the browsers. As the Examiner will note from paragraph [0014] of the instant published application 2002/0165909, session tracking objects constitute information about the requests for web pages by the browsers and this information identifies each of the browsers. Again, this distinction is not without a difference because the instant invention enables session tracking of the session objects in order to detect changes in demand more rapidly.

The Examiner also identifies col. 16, lines 25-32 as disclosing the analyzing step. Applicants disagree. The noted language of ASAI merely states the following:

> In FIG. 6, the number of distribution streams currently being distributed by the cache server 10_2 is 120+1000=1120, and the number of distribution streams of the streaming data stored in both cache servers 10_2 and 10_3 is 120 for the cache server 10_2 and 500 for the cache server 10_3. Therefore, 620 is compared with MAX for the determination in step S122, while 1120 is compared with {((n-1)/n) x MAX} for the determination in step S123.

Applicants also fail to understand how such language can be interpreted to disclose analyzing the identifications of web pages requested by the browsers to determine caching priorities for the web server, if an HTTP session object does not exist for one of browsers which requested one of the web pages, creating with the web server

an HTTP session object for the browser. At the very least, the Examiner should specifically explain how such language can be so interpreted.

The Examiner acknowledges that ASAI fails to disclose or suggest if an HTTP session object does not exist for one of browsers which requested one of the web pages, creating with the web server an HTTP session object for the browser. However, the Examiner explains that this feature is taught on paragraphs [0332] and [0335] of KNOUSE. Applicants disagree. The noted language of KNOUSE merely states the following:

> [0332] In step 3110, an authentication scheme object (ObAuthenticationScheme) is created. The constructor for the authentication scheme object is passed the resource request object. In step 3112, a user session object (ObUserSesssion) is created. The constructor for the user session object is passed the session token. The key for decrypting the session token is fetched from the directory server (if it is not already local), the session token is decrypted and the contents of the session token is stored in the new object. In step 3114, the system determines whether the cookie is valid. There are many ways for determining if the cookie is valid. In one example, the application can request information from the user session object such as the start time or last use time to determine whether the session is still valid. If the cookie was not valid, then user must be authenticated and authorized in step 3116. If the cookie is valid, then in step 3118 the application requests the authentication level from the authentication scheme object. For example, the application can call the get Level( ) method from the ObAuthenticationScheme object. This authentication level pertains to the authentication rule or policy stored in the directory server for the resource. In other embodiments, the different portions of the authentication scheme or all portions of the authentication scheme can be reported to the application by the API. In step 3120, the application requests the authentication level from the session object; for example, calling the getLevel( ) method from the ObUserSession object. This level information is a number originally stored (and encrypted) in the cookie. The ObUserSession provides the level in an unencrypted form.
>
> [0335] If the resource is not protected the application will allow the user to access the requested resource. If the resource is protected, the application accesses the authentication scheme in step 3208. One means for determining the resource authentication scheme is to use the various methods of the ObAuthenticationScheme class, described above. In step 3210, the application requests authentication credentials from the user and stores the credentials in a table in step 3212. The authentication credentials can be any data needed to authenticate. For example, a basic authentication credentials may include a username and a password. The exact type of credentials not important to the present invention. In one embodiment, the credentials is stored in a hash table.

> In step 3114, a user session object is created, if it has not already been created. The user session object is passed the resource request object and credentials stored in the table. The constructor of the user session object uses the resource request object and the credentials to authenticate the user in step 3216. The process of authentication is performed by the Access Server as described above. If the user is not properly authenticated (step 3222), then the application will send a response to a web browser in step 322 that the authentication failed and the user will not be given access to the resource.

Applicants also fail to understand how such language can be interpreted to disclose if an HTTP session object does not exist for one of browsers which requested one of the web pages, creating with the web server an HTTP session object for the browser. Again, at the very least, the Examiner should specifically explain how such language can be so interpreted.

GLANCE does not cure the deficiencies of ASAI and KNOUSE. As explained above, GLANCE merely discloses a system and method of caching based on a recommender system. The disclosed system employees a democratic caching generally shown by reference numeral 10. A recommender system 16 provides value information pertaining to items to be stored in cache 24 based on user input (col. 4, liens 43-53) that includes implicit site recommendations (col. 5, lines 24-55) and explicit URL recommendations (col. 5, lines 65 et seq.). GLANCE, like ASAI and KNOUSE, does not disclose or suggest, among other things, associating session tracking objects with browsers that access a web server, much less, that the session tracking objects include identifications of web pages requested by the browsers.

As GLANCE fails to cure the deficiencies of ASAI and KNOUSE, there can be no motivation to combine these references. Furthermore, even if ASAI, KNOUSE and GLANCE were properly combinable, the combination would not result in the invention as recited in at least claim 20.

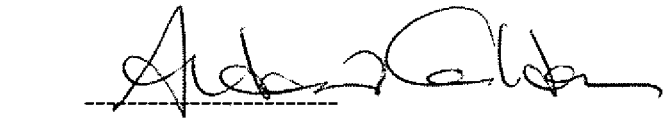
Because, there is no suggestion or disclosure in ASAI, KNOUSE and GLANCE separately or in any proper combination that render obvious the features of the present claimed invention, the Examiner is respectfully requested to withdraw the rejection under 35 U.S.C. § 103.

Thus, claim 20 Is allowable over the combination of ASAI, KNOUSE and GLANCE.

CONCLUSIONS

In view of the foregoing amendments and remarks, Applicants submit that all of the rejections have been overcome, and that the claims are patentably distinct from the prior art of record and in condition for allowance. The Examiner is respectfully requested to pass the above application to issue, and to contact the undersigned at the telephone number listed below, if needed. Applicants hereby make a written conditional petition for extension of time, if required. Please charge any deficiencies in fees and credit any overpayment of fees to **IBM Deposit Account No. 09-0457 (Endicott).**

Respectfully submitted,
Michael Christopher MARTIN, *et al.*



Andrew M. Calderon
Reg. No. 38,093

November 20, 2006
GREENBLUM & BERNSTEIN, P.L.C.
1950 Roland Clarke Place
Reston, VA 20191
703-716-1191